Qin *EURASIP Journal on Embedded Systems* (2016) 2016:4
DOI 10.1186/s13639-016-0027-9

EURASIP Journal on Embedded Systems
a SpringerOpen Journal

RESEARCH Open Access

Research of multiple-instance learning for target recognition and tracking

Jiang Qin

Abstract

Target recognition and tracking is a hot research in image and video processing and is widely used in motion analysis, behavior recognition, and so on. In this paper, we studied target recognition and tracking in a series of images, and our approach is based on the multiple-instance learning technique. Firstly, we present a general target tracking framework. Within the proposed framework, we use image frames to generate positive and negative samples to train a classifier and use the classifier to differentiate target from its background. We use a set of weak classifiers to construct a strong classifier. The experiments show that the proposed approach has better precision and recall on two public datasets than related works.

Keywords: Image process, Target recognition, Target tracking, Multiple-instance learning

1 Introduction

Target recognition and tracking is applied in many fields, such as motion analysis [1] and behavior recognition [2]. However, occlusion, similar background, lighting, surface, and etc. pose great challenges for target recognition and tracking, which will make target shift or even tracking fail [3]. Appearance model-based tracking algorithms [4,5] represent targets with scale-invariant feature transformation or histogram of oriented gradient, but these features cannot reflect the basis of targets, and mismatches usually appear in the process of tracking. Moreover, complex appearance models lead to very high computation.

The combination of appearance model and traditional machine learning techniques consumes target tracking as a binary classification problem [6,7], and this method can utilize background information effectively and thus can improve the effectiveness of tracking. However, as there are not enough training data to the classification model, the recognition ability of target is very low and thus misclassification usually occurs. Deep learning is a hot research in image and visual processing. According to construct deep non-linear network model [8,9], the essential features of images can be learned with the constructed model, and then, the classification accuracy is improved.

Flock of tracker [10] combines local trackers with global motion model and can handle the problem of occlusion and local changes of non-rigid targets. Cell flock of tracker [11] tracks targets with the selected optimal local tracker and thus can handle the problem of target shifting and is more robust in target tracking.

Multiple-instance learning is first proposed by Dietterich et al. [12], and it is the fourth machine learning technique besides supervised learning, unsupervised learning, and reinforcement learning. Zhang et al. [13] propose to embed multiple-instance learning into the AnyBoost algorithm framework and construct the MILBoost classifier for target detection. Babenko et al. [14] use multiple-instance learning for target tracking, which gets a good tracking effectiveness, so multiple-instance learning becomes a hot research in target tracking. Zeisl et al. [15] apply the semi-supervised multiple-instance learning for target tracking, in which the target and background of the first frame is assumed to be tagged sample, and targets of the subsequent frames are assumed untagged samples. When the first frame comes, the tagged sample and untagged samples, which are tracked correctly, are priors for the following frame, and this improves the stability of target tracking [16]. In addition, Babenko et al. [17] has analyzed the visual tracking with online multiple-instance learning, but

Correspondence: jiangqinxz@sina.com
Department of Computer Science, Yunyang Teachers College, Shiyan 442000, China

they aim to track the predefined target, and our method can recognize any target from its background.

However, the original multiple-instance learning has the weaknesses of low classification effectiveness and real-time ability. In order to handle these weaknesses, we propose a new weak classifier, which assigns different positive samples, different weights and assigns, different weak classifiers, and different weights. In addition, we propose a strong classifier to improve the accuracy and real-time ability of target tracking.

The rest of the paper is organized as follows. In Section 2, we present our proposed target tracking algorithm based on multiple-instance learning. Experiments and conclusion are given in Sections 3and4, respectively.

2 Multiple-instance learning target tracking algorithm

The flowchart of a tracking system is in Fig. 1, where we use all previous frames as training data to train a classifier and use this classifier to classify the$t+1$-th frame; once the$t+1$-th frame is classified, we add it into the training data for future prediction. The classifier evolves as time goes on.

2.1 Selection of positive and negative samples

During the process of traditional target tracking, the target is usually one candidate object. When the target changes a lot or is occluded, the tracking frame shifts easily. Taking the limit of single candidate target, we consider multiple candidate targets. Here, we consider the target as positive sample and consider the background as negative samples. The samples including both positive and negative samples are denoted as X. Let the location of a sample be$l_t$at time$t$, then the category of sample is$y \in \{0,1\}$, where$y=1$, if$X$is the target, and$y=0$, if X is the background.Let the location of the target be l^*_{t-1} at time$t-1$, then the sample set that is waited for classification at time$t$ is

$$X^s = \{X | \|l(X) - l^*_{t-1}\| < s\}, \tag{1}$$

where $l(X)$ is the location of sample X and s is the searching radius.

In order to acquire the location l^*_t of the target at time t, compute the probability $p(y=1)$ that all samples X is a positive sample. Let the probability that the target occurs in a cycle region with radius$s$be uniform, then we have

$$p(l^*_t | l^*_{t-1}) = \begin{cases} 1 & \|l(X) - l^*_{t-1}\| < s \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

Then, the new location of the target is

$$l^*_t = l\left(\arg\max_{X \in X^s} p(y=1|X) \right). \tag{3}$$

When the new location is calculated out, we need to select new positive and negative samples to update the classifier. While selecting the positive samples, the positive sample set X^+contains N samples, which is a cycle with l^*_t as its center, radius α, that is

$$X^+ = \{X_{1i} | \|l(X) - l^*_t\| < \alpha\}. \tag{4}$$

The negative sample set X^-contains L samples, which is a cirque with l^*_t as its center, radius from β to γ, that is

$$X^- = \{X_{0i} | \beta < \|l(X) - l^*_t\| < \gamma\}. \tag{5}$$

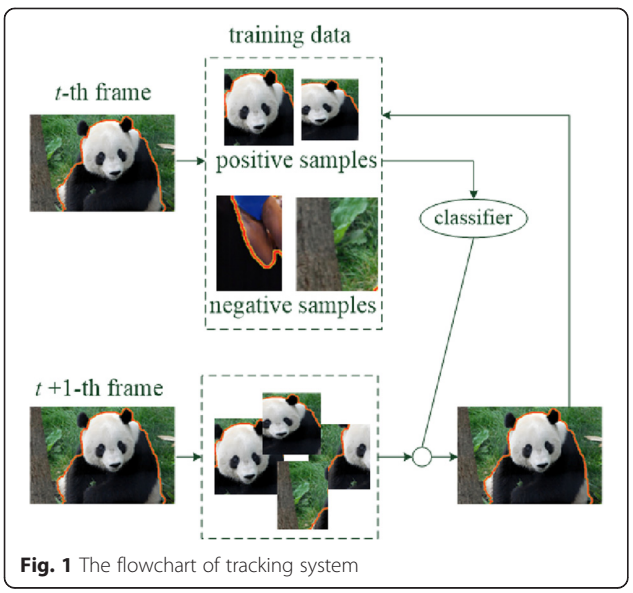

Fig. 1 The flowchart of tracking system

2.2 Training a classifier

While training the classifier, we use the selected positive and negative sample set, X^+ and X^-, and then, the probability that a sample is a positive sample is as follows [14]:

$$\begin{aligned} p(y=1|X) &= \frac{e^{H(X)}}{e^{H(X)} + e^{-H(X)}} \\ &= 0.5 \tanh(H(X)) + 0.5, \end{aligned} \tag{6}$$

where $\tanh(z) = \frac{e^{H(X)} - e^{-H(X)}}{e^{H(X)} + e^{-H(X)}}$, $H(X)$ is a strong classifier of the samples and consists of K weak classifiers.

The definition of $H(X)$ is in the following equation:

$$H(X) = \sum_{k=1}^{K} \lambda_k h_k(X), \tag{7}$$

where $h_k(X)$ is the kth weak classifier and λ_k is its weight. The weak classifiers are selected according to

their classification ability. If a weak classifier is good at classification, then we give it a big weight; otherwise, we give it a small weight. Let $\lambda_k = e^{\frac{1-k}{K}}$, then the weak classifier is selected from the set of weak classifier set Φ, where $\Phi = \{h_1, \ldots, h_M\}$ and $M > K$. The weak classifier set is generated with the following method: let $h_k = \log\left(\frac{p(y=1|f_k(X))}{p(y=0|f_k(X))}\right)$, where $f_k(X)$ is the Haar-like feature [18]; let $p(y=0) = p(y=1)$, then, with the Bayes rule, we can have $h_k = \log\left(\frac{p(f_k(X)|y=1)}{p(f_k(X)|y=0)}\right)$, where $p(f_k(X)|y=1)$ and $p(f_k(X)|y=0)$ conform to the Gaussian distribution [19], that is

$$p(f_k(X)|y=1) \sim N(\mu_1, \sigma_1), \tag{8}$$

$$p(f_k(X)|y=0) \sim N(\mu_0, \sigma_0), \tag{9}$$

where μ_1, σ_1, μ_0, and σ_0 are expectations and variances of the two Gaussian distributions.

During the training of the classifier, we use the gradient descent method, and the iterations of μ_i and σ_i are as follows:

$$\mu_i = \eta\mu_i + (1-\eta)\frac{1}{N}\sum_{j|y=1} f(X_j), \tag{10}$$

$$\sigma_i = \eta\sigma_i + (1-\eta)\sqrt{\frac{1}{N}\sum_{j|y=1}(f(X_j)-\mu_i)^2}, \tag{11}$$

where $i = 0, 1$, η is the learning coefficient.

2.3 Selecting weak classifiers

As we can see from Eq. 7, target tracking needs to use a set Φ of K weak classifiers, and then, the rule for the selection of weak classifiers is to assure an optimal strong classifier [20]. Babenko et al. [14] propose to ascertain weak classifier h by maximizing the log-likelihood function with both positive and negative sample sets, that is

$$h_k = \arg\max_{h\in\Phi} L(H_{k-1} + \lambda_k h), \tag{12}$$

where $L(H)$ is computed as follows:

$$L(H) = \sum_{s=0}^{1}(y_s \log(p(y=1|X^+)) + (1-y_s)\log(p(y=0|X^-))), \tag{13}$$

where $p(y=1|X^+) = \sum_{j=1}^{N-1} w_j p(y=1|X_{1j})$. As there exists similarity between positive sample and negative sample, we define the similar coefficient as follows:

$$w_j = \frac{1}{c}e^{-|l(X_{1j})-l((X_{10}))|}, \tag{14}$$

where c is the normalization constant.

With the same reason, we can have

$$\begin{aligned} p(y=0|X^-) &= \sum_{j=N}^{N+L-1} w_j' p(y=0|X_{0j}) \\ &= w\sum_{j=N}^{N+L-1}(1-p(y=1|X_{1j})). \end{aligned} \tag{15}$$

In Eq. 15, the similarities between negative samples are small, so we let w be constant.

Computing h with Eq. 12consumes a lot of computing resources, so we use a more efficient approach. Unwrapping $L(H_{k-1} + \lambda_k h)$ with the first-order Taylor formula, we have

$$L(H_{k-1} + \lambda_k h) \approx L(H_{k-1}) + < \lambda_k h, \nabla L(H) > |_{H=H_{k-1}}, \tag{16}$$

where $< \lambda_k h, \nabla L(H) > = \frac{\lambda_k}{N+L}\sum_{j=0}^{N+L-1} h(x_{ij})\nabla L(H)(X_{ij})$.

$$\begin{aligned} \nabla L(H)(X_{ij}) &= \frac{\partial L(H+\theta 1_{X_{ij}})}{\partial\theta}|_{\theta=0} \\ &= \frac{\partial}{\partial\theta}\sum_{s=0}^{1}\Big(y_s \log\Big(\sum_{j=0}^{N-1} w_j(0.5\tanh(H(X_{1m})+\theta 1_{X_{ij}})+0.5)\Big) \\ &\quad +(1-y_s)\log\Big(\sum_{j=N}^{N+L-1}(1-(0.5\tanh(H(X_{0m})+\theta 1_{X_{ij}})+0.5)) \\ &\quad +\log\left(c^{-y_s w^{1-\gamma_s}}\right)\Big)|_{\theta=0} = \frac{\partial}{\partial\theta}\sum_{s=0}^{1}\Big(y_s\log\Big(\sum_{j=0}^{N-1} w_j\Big(0.5\tanh\Big(H(X_{1m}) \\ &\quad +\theta 1_{X_{ij}}+0.5+(1-y_s)\log\Big(\sum_{j=N}^{N+L-1}\Big(1-\Big(0.5\tanh(H(X_{0m})+\theta 1_{X_{ij}}) \\ &\quad +0.5\Big)\Big)|_{\theta=0} = y_i\frac{w_j(1-\tanh^2(H(X_{0m})))}{\sum_{m=0}^{N-1} w_j(\tanh(H(X_{0m}))+1)} \\ &\quad -(1-y_i)\frac{(1-\tanh^2(H(X_{0m})))}{\sum_{m=N}^{N+L-1}(1-\tanh(H(X_{0m})))}, \end{aligned}$$

where $y_i = i$ and $i = 0, 1$.

$L(H_{k-1})$ is already known, so in order to compute the maximum of $L(H_{k-1} + \lambda_k h)$, we only need to compute the maximum of $< \lambda_k h, \nabla L(H) > |_{H=H_{k-1}}$; then, the Eq. 12 can be rewrote as follows:

$$h_k = \arg\max_{h\in\Phi} < \lambda_k h, \nabla L(H) > . \tag{17}$$

In the MIL algorithm proposed by Babenko et al. [14], it needs to maximize Eq. 13, and this would compute additional M probabilities belonging positive or negative set for each sample, so the computing complexity is very high. In this paper, we propose an algorithm for computing $H(X) = \sum_{k=1}^{K}\lambda_k h_k(X)$, and the algorithm is in algorithm 1. According to the first frame of a video, we find the target to be tracked and generate positive and negative sample set $\{X^+, X^-\}$, where $X^+ = \{X_{1j}, y_1 = 1, j = 0, 1, \ldots, N-1\}$, and $X^- = \{X_{0j}, y_0 = 1, j = N, 1, \ldots, N+L-1\}$.

Next, according to Eqs. 8 and 9, we compute $p(f(X_{1j})|y = 1)$ and $p(f(X_{0j})|y = 0)$ and then compute h_k for k from 1 to M to generate weak classifier set $\Phi = \{h_1, \ldots, h_M\}$.

Algorithm 1. Computation of $H(X)$

1. Initialize $H_0(X_{ij}) = 0$;
2. for k from 1 to K
3. let $L_m = 0$, $m = 1,...,M$, compute $\nabla L(H)(X_{ij})$;
4. for m from 1 to M
5. for i from 0 to 1
6. for j from 0 to $N + L - 1$
7. $L_m = L_m + h_m(X_{ij})\nabla L(H)(X_{ij})|_{H=H_{K-1}}$;
8. end for
9. end for
10. end for
11. $h_k(x_{ij}) = \arg\max_{h\in\Phi}(L_m)$;
12. end for
13. compute $p(y = 1|X)$ with equation6;
14. compute $H(X) = \sum_{k=1}^{K} \lambda_k h_k(X)$;
15. return $H(X)$;

3 Experiments

3.1 Experimental setup

In the experiments, we use iCoseg [21] and MSRC [22], the two public datasets. The iCoseg dataset consists a series of related images for each object. For example, an athlete moves on a horizontal bar. The MSRC dataset monitors an environment in a forest. In this dataset, a panda occurs and disappears in the camera. We test target recognition and tracking in these two scenes.

The baseline algorithms are MIL [14], OAB [23], and SBT [6]. The MIL algorithm is a classical multiple-instance learning approach for target tracking. The OAB algorithm is a boosting approach for target classification in image series. The SBT algorithm is a semi-supervised machine learning approach, and it uses massive untagged data to improve the accuracy of classification.

3.2 Experimental results

While evaluating the performance of the proposed algorithm, we use precision and recall two metrics. Here, we use "Jumping" to represent a woman moving on a horizontal bar and 'panda' to represent a panda appearing in a camera.

Firstly, we compare the precision of the four algorithms on both two datasets, and the result is in Fig. 2. As we can see from the figure, the OAB and SBT algorithms have better precisions in Jumping than they are in the panda dataset. Moreover, the MIL algorithm has better precision in the panda dataset than it is in the Jumping dataset. The above observation concludes that different tracking algorithms would have different precision in different scenes. However, as we use multiple-instance learning while classifying target from its background, it has the best precision in both of the two dataset.

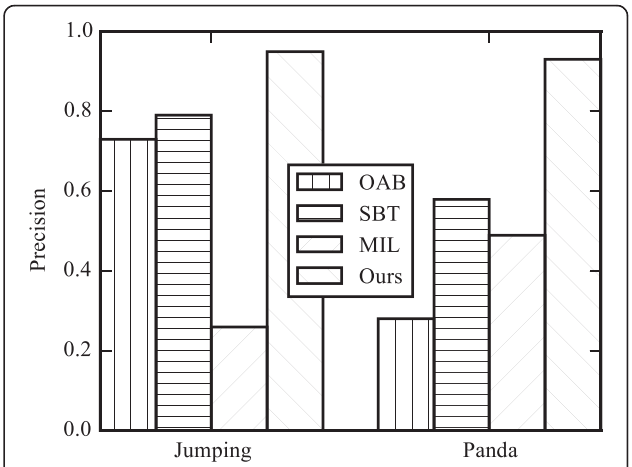

Fig. 2 Comparison of precision

Secondly, we compare the recall of the four algorithms on both of the two datasets, and the result is in Fig. 3. As we can see from the figure, the OAB and SBT algorithms have lower recalls in Jumping than they are in the panda dataset. Moreover, the MIL algorithm has better recall in the Jumping dataset than it is in the panda dataset. The above observation also concludes that different tracking algorithms would have different recalls in different scenes. However, as we use multiple-instance learning while classifying target from its background, it has the best recall in both of the two dataset.

Next, we illustrate the target recognition results on these two scenes, and the results are in Fig. 4. The

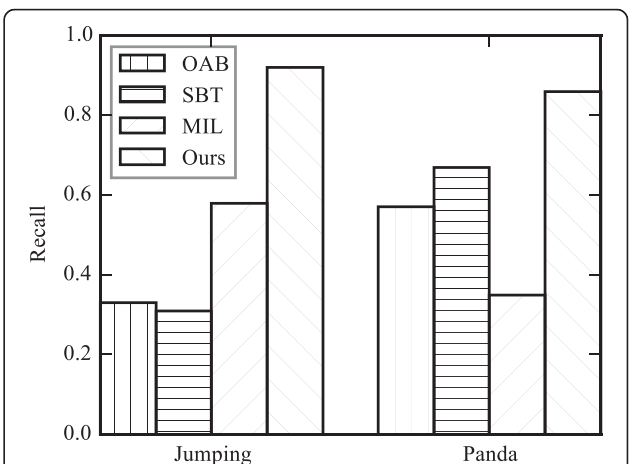

Fig. 3 Comparison of recall

Fig. 4 Illustration of target recognition results in image series

images in the first line capture the panda. Whenever the panda sits down, walks, or crosses a river, it can be easily recognized. Even some part of the panda is not in the images, the panda can also be recognized. The images in the second line illustrate the recognition of a woman while she is moving on a horizontal bar. In this scene, the backgrounds in the images are almost the same, and the woman does different actions. This situation is much easier than the last one, and classification accuracy can be assured. In this dataset, even though some parts of the woman are occluded, the woman can also be recognized clearly.

Finally, we compare the performances of executing time and memory usage of the algorithms on the two datasets. Figure 5 illustrates the executing time comparison, and from the figure, we can see that our proposed algorithm consumes the least executing time under both datasets, the OAB algorithm is the second least, and the other two algorithms take longer executing time. While comparing SBT and MIL, the MIL algorithm takes the longest executing time under the jumping dataset and the SBT algorithm takes the longest executing time under the panda dataset. Figure 6 illustrates the memory usage comparison of the algorithms under both datasets. From this figure, we can see that our proposed algorithm consumes the least memory usage while recognizing and tracking targets under the two datasets and the OAB algorithm consumes the second least memory on both datasets. In addition, for SBT and MIL algorithms, SBT needs more memory than MIL under the jumping dataset and MIL needs more memory than SBT under the panda dataset.

4 Conclusions

In this paper, we studied target recognition and tracking in a series of images, and our approach is based on the multiple-instance learning technique. In the target tracking framework, we use image frames to generate positive and negative samples to train a classifier, and use the classifier to differentiate target from its background. We use a set of weak classifiers to construct a strong classifier. The experiments show that the proposed approach has better precision and recall on two public datasets than related works.

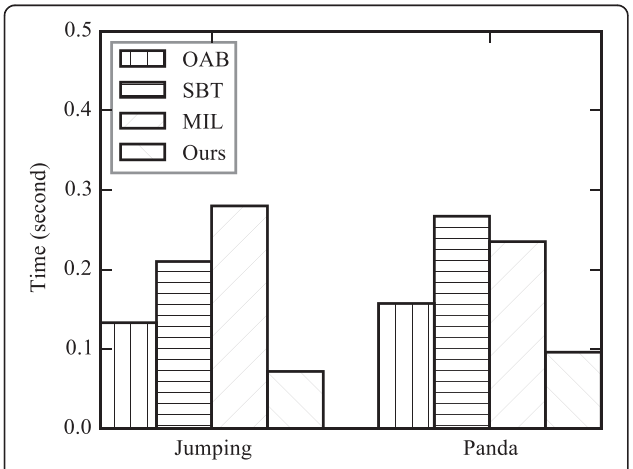

Fig. 5 Comparison of executing time

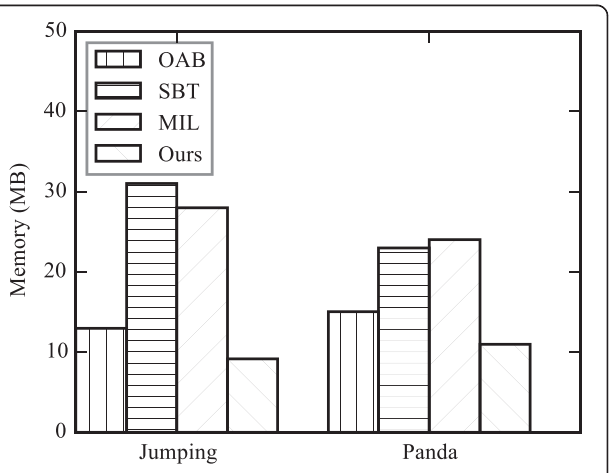

Fig. 6 Comparison of memory

Competing interests
The author declares no competing interests.

Acknowledgements
This work was financially supported by the Science and Technology Research Program for the Education Department of Hubei province of China (Q20156002).

Received: 2 December 2015 Accepted: 2 March 2016
Published online: 15 March 2016

References

1. L Chen, H Wei, J Ferryman, A survey of human motion analysis using depth imagery. Pattern Recogn Lett **34**(15), 1995–2006 (2013)
2. OP Popoola, K Wang, Video-based abnormal human behavior recognition—a review. IEEE Trans Syst Man Cybern Part C Appl Rev **42**(6), 865–878 (2012)
3. A Milan, K Schindler, S Roth, Challenges of ground truth evaluation of multi-target tracking, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013, pp. 735–742
4. X Jia, H Lu, MH Yang, Visual tracking via adaptive structural local sparse appearance model, in *IEEE Conference on Computer vision and pattern recognition (CVPR)*, 2012, pp. 1822–1829
5. S Zhang, H Yao, X Sun et al., Robust visual tracking using an effective appearance model based on sparse coding. ACM Trans Intell Syst Technol **3**(3), 43 (2012)
6. H Grabner, C Leistner, H Bischof, Semi-supervised on-line boosting for robust tracking, in *Computer Vision–ECCV*(Springer, Berlin Heidelberg, 2008), pp. 234–247
7. Z Kalal, J Matas, K Mikolajczyk, Pn learning: bootstrapping binary classifiers by structural constraints, in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, 2010, pp. 49–56
8. M Denil, L Bazzani, H Larochelle et al., Learning where to attend with deep architectures for image tracking. Neural Comput **24**(8), 2151–2184 (2012)
9. S Zhang, H Yao, X Sun et al., Sparse coding based visual tracking: review and experimental comparison. Pattern Recogn **46**(7), 1772–1788 (2013)
10. V Tomas, M Jiri, *Robustifying the flock of trackers*(Proceedings of Computer Vision Winter Workshop, Graz, Austria, 2011), pp. 91–97
11. ME Maresca, A Petrosino, Clustering local motion estimates for robust and efficient object tracking. in *Computer Vision-ECCV 2014 Workshops*. Springer International Publishing, 2014, pp. 244–253
12. TG Dietterich, RH Lathrop, T Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles. Artif Intell **89**(1), 31–71 (1997)
13. C Zhang, JC Platt, PA Viola, Multiple instance boosting for object detection, in*Advances in neural information processing systems*, 2005, pp. 1417–1424
14. B Babenko, MH Yang, S Belongie, Robust object tracking with online multiple instance learning. IEEE Trans Pattern Anal Mach Intell **33**(8), 1619–1632 (2011)
15. B Zeisl, C Leistner, A Saffari et al., On-line semi-supervised multiple-instance boosting, in*2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1879–1879
16. Z Wang, S Yoon, S Xie J et al., Visual tracking with semi-supervised online weighted multiple instance learning. Vis. Comput. 2015, pp. 1–14.
17. B Babenko, MH Yang, S Belongie, Visual tracking with online multiple instance learning, in*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 983–990
18. R Lienhart, J Maydt, An extended set of haar-like features for rapid object detection, in*International Conference on Image Processing*, 2002. 1: I-900-I-903 vol. 1
19. J Gao, H Ling, W Hu et al., Transfer learning based visual tracking with gaussian processes regression. in *Computer Vision–ECCV 2014*. Springer International Publishing, 2014, pp. 188–203
20. B Ma, J Shen, Y Liu et al., Visual tracking using strong classifier and structural local sparse descriptors. IEEE Trans Multimedia **17**(10), 1818–1828 (2015)
21. D Batra, A Kowdle, D Parikh et al., Icoseg: interactive co-segmentation with intelligent scribble guidance, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3169–3176
22. JC Rubio, J Serrat, A López et al., Unsupervised co-segmentation through region matching, in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 749–756
23. H Grabner, M Grabner, H Bischof, Real-time tracking via on-line boosting, in *British Machine Vision Conference*, 2006, pp. 47–56